IN THE CLAIMS:

Please cancel claims 30-32 without prejudice and amend the claims as follows:

1. (Original) A method for serving data comprising the steps of:

maintaining an incomplete version of an object at a server;

maintaining at least one fragment at the server;

in response to a request for the object from a client, sending to the client the incomplete version of the object, an identifier for a fragment comprising a portion of the object, and a position for the fragment within the object;

after receiving the incomplete version of the object, the identifier, and the position, requesting, by the client, the fragment from the server using the identifier; and

constructing the object by including the fragment in the incomplete version of the object in a location specified by the position.

2. (Original) The method of claim 1, wherein the client comprises a cache.

3. (Original) The method of claim 1, wherein the object comprises a Web page.

4. (Original) The method of claim 1, wherein the step of constructing further comprises the steps of:

determining whether a depth of inclusion relationships in the object exceeds a threshold; and

in response to the depth exceeding the threshold, abandoning constructing the object.

5. (Original) The method of claim 4, further comprising the step of increasing the threshold until the object is constructed.

6. (Original) The method of claim 1, further comprising the step of maintaining a list of fragment identifiers corresponding to inclusion relationships.

7. (Original) The method of claim 6, wherein the list of fragment identifiers are included in a hash table, and further including the step of examining the hash table each time the inclusion relationships change.

8. (Original) The method of claim 7, wherein the inclusion relationships are stored in a list, and further comprising the step of examining the list of inclusion relationships when the list of inclusion relationships increases by a new fragment to see if the new fragment has already been encountered in the list of inclusion relationships.

9. (Original) The method of claim 8, wherein if the new fragment has already been encountered, concluding that a cycle exists; otherwise, adding the new fragment identifier to the hash table.

10. (Original) The method of claim 9, further comprising the step of after completing construction with the new fragment, removing the new fragment.

11. (Original) The method of claim 6, further comprising representing fragments by vertices in a directed graph, and inclusion relationships by directed edges in the graph, the method further comprising the step of determining if there is a cycle in a list of inclusion relationships.

12. (Original) The method as recited in claim 11, wherein the step of determining includes traversing a list of inclusion relationships using two pointers.

13. (Original) The method as recited in claim 12, wherein the two pointers include a first pointer which traverses one fragment at a time, and a second pointer which traverses two fragments at a time, and if the two pointers meet in the graph, determining that a cycle has been detected.

14. (Original) A program storage device readable by machine, tangibly embodying a program of instructions executable by the machine to perform method steps as recited in claim 1.

15. (Original) In a system including at least one server and at least one client, a method for serving data comprising the steps of:

maintaining at the at least one server, an incomplete version of an object;

maintaining at the at least one server, at least one fragment;

in response to a request for the object from a client, the server sending to the client the incomplete version of the object, at least one identifier for a fragment comprising a portion of the object, and at least one position for the fragment within the object;

the client receiving the incomplete version of the object, the at least one identifier, and the at least one position;

the client requesting the at least one fragment from the at least one server using the at least one identifier; and

the client constructing the object by including the at least one fragment in the incomplete version of the object in a location specified by the at least one position.

16. (Original) The method of claim 15, wherein the at least one client comprises at least one cache.

17. (Original) The method of claim 15, wherein the object comprises a Web page.

18. (Original) The method of claim 15, wherein the step of constructing further comprises the steps of:

determining whether a depth of inclusion relationships in the object exceeds a threshold; and

in response to the depth exceeding the threshold, abandoning constructing the object.

19. (Original) The method of claim 18, further comprising the step of increasing the threshold until the object is constructed.

20. (Original) The method of claim 15, further comprising the step of maintaining a list of fragment identifiers corresponding to inclusion relationships.

21. (Original) The method of claim 20, wherein the list of fragment identifiers are included in a hash table, and further including the step of examining the hash table each time the inclusion relationships change.

23. (Currently Amended) The method of claim 22 20, wherein the inclusion relationships are stored in a list, and further comprising the step of examining the list of inclusion relationships when the list of inclusion relationships increases by a new fragment to see if the new fragment has already been encountered in the list of inclusion relationships.

24. (Original) The method of claim 23, wherein if the new fragment has already been encountered, concluding that a cycle exists; otherwise, adding the new fragment identifier to the hash table.

25. (Original) The method of claim 24, further comprising the step of after completing construction with the new fragment, removing the new fragment.

26. (Currently Amended) The method of claim 20 15, further comprising representing fragments by vertices in a directed graph, and inclusion relationships by directed edges in the graph, the method further comprising the step of determining if there is a cycle in a list of inclusion relationships.

27. (Original) The method as recited in claim 26, wherein the step of determining includes traversing a list of inclusion relationships using two pointers.

28. (Original) The method as recited in claim 27, wherein the two pointers include a first pointer which traverses one fragment at a time, and a second pointer which traverses two fragments at a time, and if the two pointers meet in the graph, determining that a cycle has been detected.

29. (Original) A program storage device readable by machine, tangibly embodying a program of instructions executable by the machine to perform method steps as recited in claim 15.

Claims 30-32 (Canceled)